Programming in Ceemac

(A primer written by someone who barely knows anything about it)

By Kay Savetz
September 12, 2021

Ceemac is a programming language for the Apple II that is geared toward making pretty, fast-moving graphics. (I believe it was only released for the Apple II. Although I have found references to Commodore 64 and IBM PC versions, I haven't seen evidence that it was released for those systems.)

When someone bought the Ceemac programming language disk from Brooke Boering back in nineteen eighty whatever, it came with a printed manual. As far as I can tell, that manual hasn't been preserved online. The Ceemac disk includes a bit of on-disk documentation, and enough magazine articles were written about Ceemac to figure out how to program in that language. But actually using the Ceemac programming/editing environment isn't intuitive or well documented. So until Boering's official manual turns up, here's a quick primer to getting around in Ceemac.

Use an Apple II+ or II+ emulator. (Things don't look right on a //e.) Boot vagabondoenterprises_ceemac_v1.dsk. At the RUN CEEMAC? (Y/N) prompt, type Y followed by RETURN.

Once it boots, you will see the Fire Organ logo. It might not be clear at first, but Fire Organ is a Ceemac program (or "score" in the parlance of the language).

Press Control-A. This will take you to the program editor, where you'll see the code for the Fire Organ score. Press Control-A again to run the program again. This way you can make changes to a program and instantly see the results of those changes.

Press Control-A once again to get back to the editor. The editor has a command mode and an editing mode. If you're familiar with the vi (or vim) editor on linux — it's a bit like that.

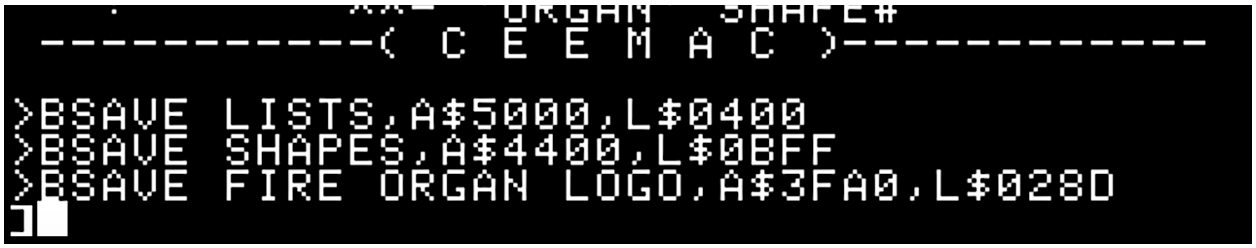
In the editor, press Return to navigate down and - (minus) to go back up. Press I to switch to edit mode by inserting lines above the current line. Press Return on a blank line to exit editing mode back to command mode.

As you add or edit code, the editor will only accept lines that are syntactically valid. If it's not, the cursor will move to the start of that line for you to fix. If you press Return without fixing it, it will delete that line. The editor is pretty fussy about syntax: for example, `COLOR = $FF` is acceptable but `COLOR=$FF` is not because of the spacing. Only one command per line. You can't invent variable names, you can only use the language's predefined ones.

Back in command mode, press D to delete a line or R to replace a line (deleting it then switching to insert mode). Type L to jump back to the top of the score. Letter O also does something, but I don't understand what — it might be an overtype mode.

It seems silly, but there's no equivalent to BASIC's "NEW" statement. When you want to write your own program from scratch (or type one in from a magazine article), start with the Fire Organ program and delete it all to start a new score: starting at the top, press Return once (you can't delete the top line) then press D 66 times to erase all lines of the program. (If you're going to do this a lot, you could save your blank program.)

To get to DOS, press control-C. To get back to the Ceemac editor, type `CALL -151` followed by Return, then `800G` followed by Return.



From DOS, you can save your current score with a BSAVE command. There's no save or load commands in the editor, so this seems to be the only way to save your work. The exact BSAVE command changes depending on the length of your score, but your current BSAVE command is helpfully shown at the bottom of the screen when you switch to DOS. It looks something like
`BSAVE MYSCORE,A$3FA0,L$028D`
That last number may be different.

(The bottom of the screen also tells you how to BSAVE your lists and shapes — those features are beyond my understanding.)

To load a different score, BLOAD it, for instance `BLOAD KZ` then `CALL -151` then `800G` to run the score. Once again, Control-A to get back to the editor to see the code.

That's all I know, but it should be enough to get you started with Ceemac. Hopefully someone will find and scan the original printed manual. I'm positive that there's stuff that I missed or got wrong here. Ceemac looks like a fun little language and I hope this primer helps people to tinker with it.

@KaySavetz